access rights. In a special case they may also contain concrete resources that do not depend on any parameters. Role types **2** can be organized hierarchically by a "subsumed" relation. If a first role type **16** subsumes a second role type **17** then the set of access rights available to an instance **18** of the first role type **16** includes those available to a corresponding instance **19** of the second role type **17**. The expression "corresponding" in this context means that both role types **16, 17** are instantiated with the same parameter values. The subsuming role type **16** must have at least the parameters of the subsumed role type **17**, but it may have more.

The role type hierarchy defines in mathematical terms a lattice structure. Trivially, the top of the lattice can include all types of access rights to all objects **4**, whereas the bottom can include the respective empty sets. Of course, lattices with non-trivial tops and bottoms can be defined. When instantiating a lattice of role types in a system, the top and bottom of the lattice need not be used for any specific role instance **3** and job position **6**.

It is the implicit assumption which leads to the notion of the role type hierarchy that the sets of generic competencies of job functions **6** and the role types **2** derived from them

1. can be structured as hierarchies by the subsumed relation, and
2. do not change very frequently.

The first assumption appears to be realistic because enterprise access control policies are often defined to reflect the hierarchical relationship built in an enterprise organization and functions. The second assumption also appears to be realistic because the job functions defined with an enterprise are stable since they are based on the enterprise business characteristics. Since the definition of job functions does not change very often, the sets of access rights to objects **4** needed for a job position **6** are not expected to change very often. It is important, that neither assumption prevents the addition of new role types **2** to the lattice nor that of new role instances **3** and job positions **6** to an enterprise.

The FIG. **3**B shows an example for the role type hierarchy within the inventive method of access control. The example shows a hierarchy of the role types **2** used in FIG. **2**B. In this example the access rights of a "second-line manager" and of a "first-line manager" subsume those of a "secretary" which in turn subsume those of a "typist". All role types subsume the role type "bank employee". As a consequence "bank employee" could be dropped from the matrix in FIG. **2**B because the corresponding competencies are covered by a membership in any of the other role types. For the same reason the "team-leader" of the "object appraisal" department does not have to be assigned the "loan specialist" role explicitly since his "team-leader" role type subsumes it.

The FIG. **4** shows the instantiation of concrete resource sets **9** and individual resources **10** from parameterized relative resource sets **8**. The parameterized relative resource sets **8** are associated to the parameterized role types **2**. The concrete resource sets **9** are derived from the parameterized relative resource sets **8** by using the parameter values provided from the subjects **6, 7** in the computer systems, e.g. provided from the job positions **6** and organization units **7** of the enterprise. The individual resources **10** are grouped to concrete resource sets **9**. For example one possible parameterized relative resource set **8** is the resource set of "printers" with a parameter "printlocation". By providing the location parameter, for example location Heidelberg, the relative resource set **8** is instantiated into the concrete resource set **9** that includes all printers at the location Heidelberg. These printers at the location Heidelberg represent the individual resources **10**.

The FIG. **5** shows an overview of the method for controlling access rights for the organizational level **20** as well as for the system level **21**. It is shown that on the system level **21** persons **5** are represented as users **22**, wherein one person **5** may have multiple user identifications, which may be derived from the role information and automatically generated (automatic registration) in the same way as the access rights are derived (automatic authorization). Furthermore, it is shown that the role instances **3** on the organization level **20** are represented by groups **23** on the system level. Furthermore, the concrete resource sets **9** are represented by the individual resources **10** on the system level **21**.

The FIG. **6** shows a preferred embodiment of a system for authorization and control of access rights as disclosed in the present invention. It is shown that capability lists **30** associated to the subjects **1** of the computer system and containing the access rights of the respective subject **1** on the objects **4** of the computer system can be derived by appropriate derivation means **32** into access control lists **31** associated to the objects **4** of the computer system and containing the access rights of the subjects **1** of the computer system on the respective object **4**. The derivation means **32** can be implemented by hardware or by software. Furthermore, it is also possible to derive capability lists **30** from existing access control lists **31**.

The FIG. **7** shows the possibility to perform a per-object review **40** with the inventive system for authorization and control of access rights. In this example the access rights may be an execute permission "X", a read permission "R" or a write permission "W". Since the inventive control system provides access control lists **31** associated with the objects **4** of the computer system it is possible to evaluate these access control lists **31** in order to determine all access rights of groups **23** within the computer system on the respective object **4**. The group **23** is the representation of an instance, i.e. a role instance **3**, of a parameterized role type **2**. The role type **2** is instantiated by at least one parameter value provided by the job position **6**. The person **5** assigned to this job position **6** has at least one user identification.

As also shown in FIG. **7**, the inventive system for authorization and control of access rights as disclosed in the present invention offers the possibility to perform a per-subject review **41**. The job position **6** to which a person **5** is assigned to is associated with a role. Associated to this role are the access rights of that role on the objects **4** of the computer system. The inventive system comprises capability lists **30** containing these access rights for each role. Furthermore, the system comprises deriving means **32** to generate new or modify existing access control lists **31** from the capability lists **30**.

What is claimed is:

1. A method for controlling access rights of at least one subject on at least one object in a computer system, wherein said subject is associated to at least one role, said method comprising the steps of:

controlling said access rights dependent on a membership of said subject to said role,

controlling said access rights dependent on a parameterized role type,

controlling said access rights dependent on at least one parameterized relative resource set,

representing said role by instantiating role instance by deriving said role instance from said role type,

said step of instantiating said role instance being based on providing a parameter value to said role type, said parameter value further characterizing said subject,